# Introduction to SQL

Class 7

# Course Overview

- Introduction to SQL
  - Databases, Tables
  - Classification of SQL – DDL, DML, DCL, TCL
    - DDL – CREATE, ALTER, DROP
    - DML – SELECT, INSERT, UPDATE, DELETE
    - DCL – GRANT, REVOKE
    - TCL – COMMIT, ROLLBACK, SAVEPOINT
  - Data types, Operators
  - Keys – Primary, Foreign, Composite, Unique, Alternate
  - Integrity Constraints – Domain Integrity Constrains, Entity Integrity Constraints, Referential Integrity Constraints
  - Joins – Outer Joins, Left Outer Joins, Right Outer Joins, Inner Joins.
  - Queries, Subqueries, Functions, Flow Control (IF, CASE, WHILE, FOR, LOOP), Stored Procedures,Stored functions
  - Views
  - Indexes, Cursors, Triggers, Events
  - Concurrency and locking (Implicit locks, explicit locks, row level locks, table level locks, database level locks)
  - Tuning SQL queries and optimizing performance
  - SQL Databases vs NoSQL Databases
  - ACID, CAP
  - How SQL databases internally works

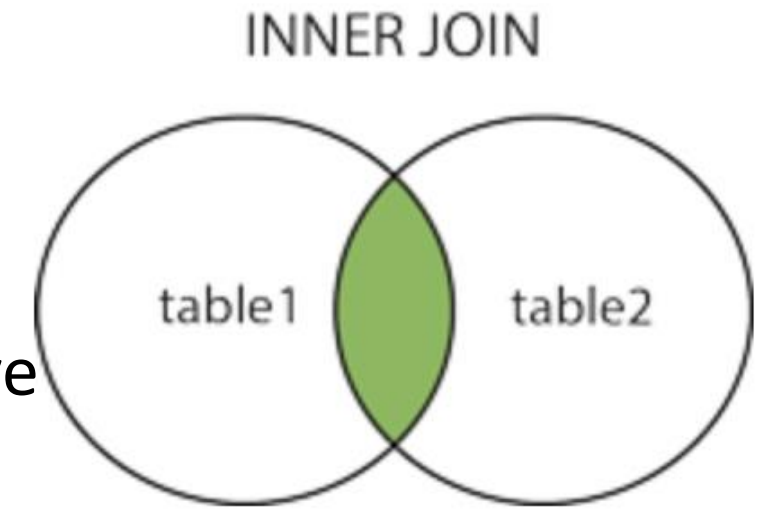
# JOINS – Why do we need joins

- Combining data from multiple tables into one result set.
- Retrieving data from multiple tables based on a common column between them.
- Merging data from tables with one-to-one, one-to-many, or many-to-many relationships.
- Consolidating data from separate tables into a single view.
- Retrieving additional data not present in one table by joining it with another.

# SQL Join

- A JOIN clause is used to combine rows from two or more tables, based on a related column between them.
- Different types of joins :
  - (INNER) JOIN: Returns records that have matching values in both tables.
  - LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table.
  - RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table.
  - FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table.

# SQL Inner Join




- The INNER JOIN keyword selects records that have matching values in both tables.
- This is also the default type of join
- SYNTAX :

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name

INNER JOIN table3
ON table2.column_name = table3.column_name;
```

# MySQL Inner Join with USING clause

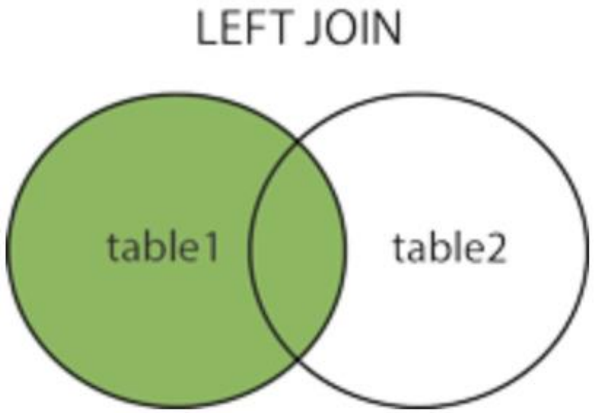
- Sometimes, the name of the columns is the same in both the tables.
- In that case, we can use a USING keyword to access the records. The following query explains it more clearly:

```
SELECT student_id, inst_name, city, technology
FROM students
INNER JOIN technologies
USING (student_id);
```

# Few questions on books tables

- What are the titles and authors' names of all the books in the library?
- How many books were written by American authors?
- Which authors have published a book in the last 10 years?
- What is the average publication year of books written by authors born after 1950?
- What are the titles of books that have the same ISBN as the book with the title 'Moby-Dick; or, The Whale'?

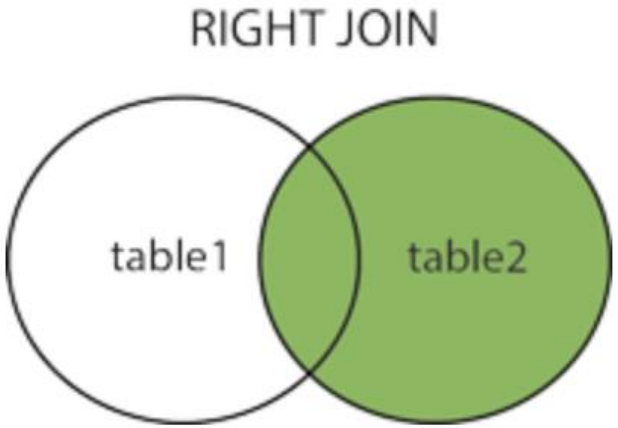
# SQL Left (Outer) Join




- Left Join clause returns all the rows from the left table and matched records from the right table or returns Null if no matching record found.
- This Join can also be called a **Left Outer Join** clause. So, Outer is the optional keyword to use with Left Join.
- SYNTAX :

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

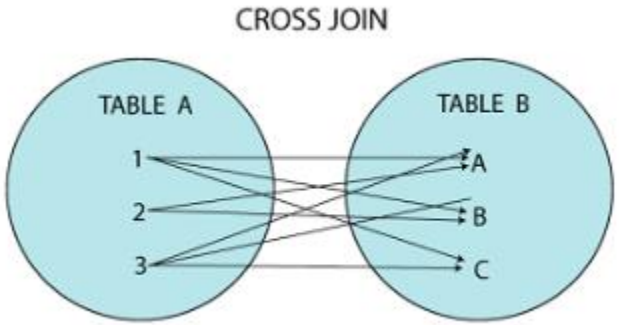
# SQL Right (Outer) Join




- The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side, if there is no match.
- SYNTAX :

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

# MySQL CROSS JOIN




- Each row is the combination of rows of both tables.
- MySQL CROSS JOIN is used to combine all possibilities of the two or more tables and returns the result that contains every row from all contributing tables.
- The CROSS JOIN is also known as CARTESIAN JOIN, which provides the Cartesian product of all associated tables.
- The Cartesian product can be explained as all rows present in the first table multiplied by all rows present in the second table.
- It is similar to the Inner Join, where the join condition is not available with this clause.

```
SELECT column-lists
FROM table1
CROSS JOIN table2;
```

# MySQL SELF JOIN

- A SELF JOIN is a join that is used to join a table with itself.
- We can perform Self Join using table aliases
- The table aliases allow us not to use the same table name twice with a single statement.
- If we use the same table name more than one time in a single query without table aliases, it will throw an error.
- The table aliases enable us to use the temporary name of the table that we are going to use in the query

Select … FROM student AS S1
INNER JOIN student AS S2;

```
SELECT s1.col_name, s2.col_name...
FROM table1 s1, table1 s2
WHERE s1.common_col_name = s2.common_col_name;
```

User Table – Table 1

| ID (Primary Key) | Name | Address |
|---|---|---|
| 1 | Sally Select | 123 Join Dr |
| 2 | Frank From | 25 Where St |

Event Table – Table 2

| User_ID (Foreign Key) | ID (Primary Key) | Action |
|---|---|---|
| 1 | A | LOGIN |
| 3 | B | VIEW PAGE |
| 4 | C | LOGIN |

Table 1

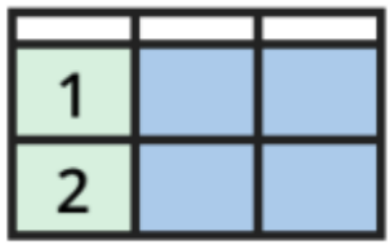
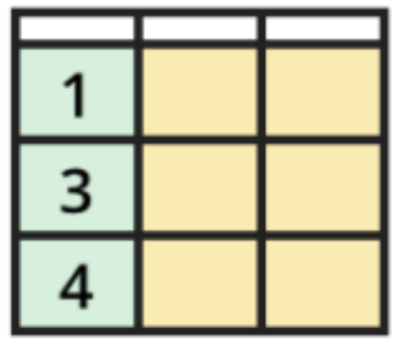



Table 2




Outer Join

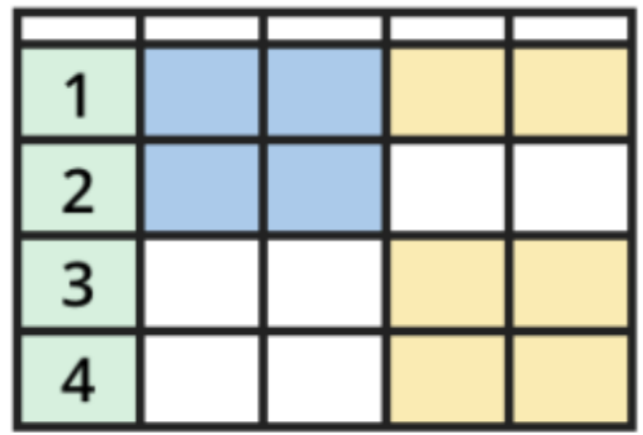


Table 1

| | | |
|---|---|---|
| 1 | | |
| 2 | | |

Table 2

| | | |
|---|---|---|
| 1 | | |
| 3 | | |
| 4 | | |

Inner Join

| | | | | |
|---|---|---|---|---|
| 1 | | | | |

Table 1

| | | |
|---|---|---|
| 1 | | |
| 2 | | |

Table 2

| | | |
|---|---|---|
| 1 | | |
| 3 | | |
| 4 | | |

Left Join

| | | | | |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | | | |

Table 1

| | | |
|---|---|---|
| 1 | | |
| 2 | | |

Table 2

| | | |
|---|---|---|
| 1 | | |
| 3 | | |
| 4 | | |

Union +

| | | |
|---|---|---|
| 1 | | |
| 2 | | |
| 1 | | |
| 3 | | |
| 4 | | |

Table 1

| | | |
|---|---|---|
| 1 | | |
| 2 | | |

Table 2

| | | |
|---|---|---|
| 1 | | |
| 3 | | |
| 4 | | |

Cross Join

| | | | | | |
|---|---|---|---|---|---|
| 1 | | | 1 | | |
| 1 | | | 3 | | |
| 1 | | | 4 | | |
| 2 | | | 1 | | |
| 2 | | | 3 | | |
| 2 | | | 4 | | |

# Combining Data Tables – SQL Joins Explained

A JOIN clause in SQL is used to combine rows from two or more tables, based on a related column between them.

Table 1

| | | |
|---|---|---|
| 1 | | |
| 2 | | |

Table 2

| | | |
|---|---|---|
| 1 | | |
| 3 | | |
| 4 | | |

Outer Join

| | | | | |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | | | |
| 3 | | | | |
| 4 | | | | |

Inner Join

| | | | | |
|---|---|---|---|---|
| 1 | | | | |

Left Join

| | | | | |
|---|---|---|---|---|
| 1 | | | | |
| 2 | | | | |

Union +

| | | |
|---|---|---|
| 1 | | |
| 2 | | |
| 1 | | |
| 3 | | |
| 4 | | |

Cross Join

| | | | | | |
|---|---|---|---|---|---|
| 1 | | | 1 | | |
| 1 | | | 3 | | |
| 1 | | | 4 | | |
| 2 | | | 1 | | |
| 2 | | | 3 | | |
| 2 | | | 4 | | |

Connect with me on LinkedIn

Please subscribe to our YouTube channel

Check out my GitHub profile

Follow me on Twitter(X)

# Thank you

- ashok-bidani/MySQL-Sakila-queries-and-joins: Queries in MySQL (github.com)
- MySQL-cheatsheet/joins.sql at master · Cheatsheet-lang/MySQL-cheatsheet (github.com)
- Cheatsheet-lang/MySQL-cheatsheet: Cheatsheet for MySQL (github.com)
- SQL-Practice/Day-3 Joins, Subqueries, Auto-Increment, Limit.txt at main · rish2408/SQL-Practice (github.com)
- shushrutsharma/18CSC303J-DBMS: All the weekly lab work of the subject 18CSC303J Data Base Management Systems. (github.com)
- 18CSC303J-DBMS/dbms11.sql at master · shushrutsharma/18CSC303J-DBMS (github.com)
- tweichle/w3resource-SQL-Exercises: Exercises to help improve SQL skills (github.com)
- Leetcode SQL
- Hackerrank SQL